SCHOOL
OF ENERGY
& POWER ENGINEERING

Алгоритмы обучения нейронных сетей

Сергей Владимирович Аксёнов,

Доцент отделения информационных технологий ИШИТР,

Томский политехнический университет

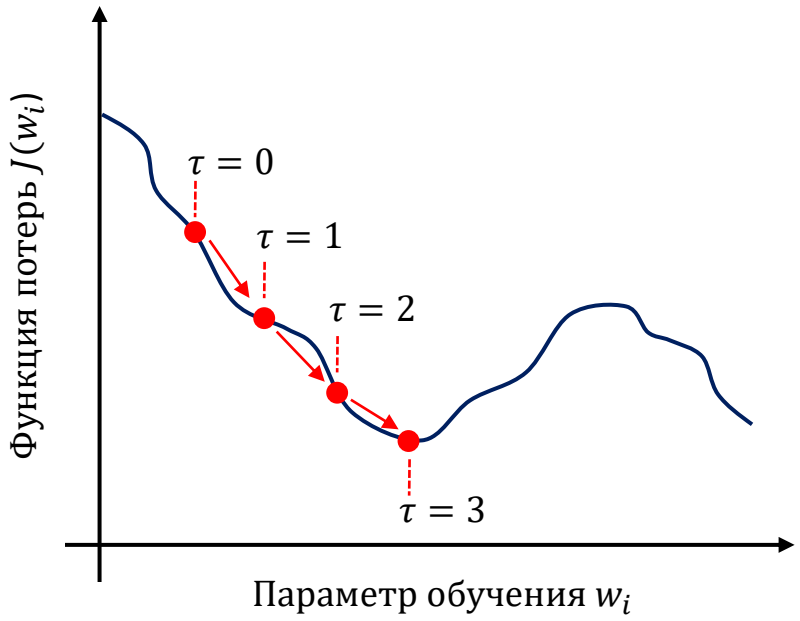
Томск-2023

Стохастический градиентный спуск

$$w_i^{(\tau+1)} = w_i^{(\tau)} - \eta^{(\tau)} \frac{\partial J}{\partial w_i^{(\tau)}}, \forall i$$

$$\frac{\partial J}{\partial w_i} = \frac{J(w_i + \varepsilon) - J(w_i - \varepsilon)}{2\varepsilon}, \varepsilon \to 0$$

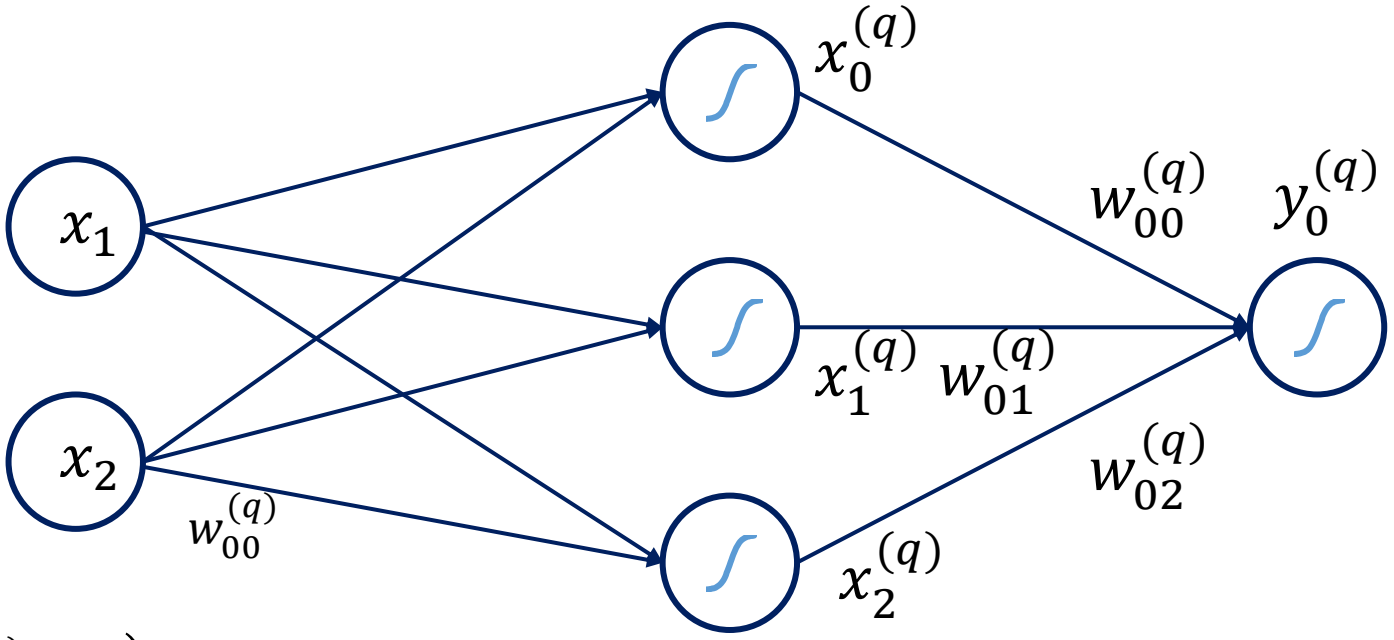
$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial y_i} \cdot \frac{\partial y}{\partial S_i} \cdot \frac{\partial S}{\partial w_i}$$



η - Скорость обучения

Обратное распространение -1

$$\frac{\partial J}{\partial w_{ij}^{(q)}} = \frac{\partial J}{\partial y_i^{(q)}} \cdot \frac{dy_i^{(q)}}{dS_i^{(q)}} \cdot \frac{\partial S_i^{(q)}}{\partial w_{ij}^{(q)}}$$



Последний слой:

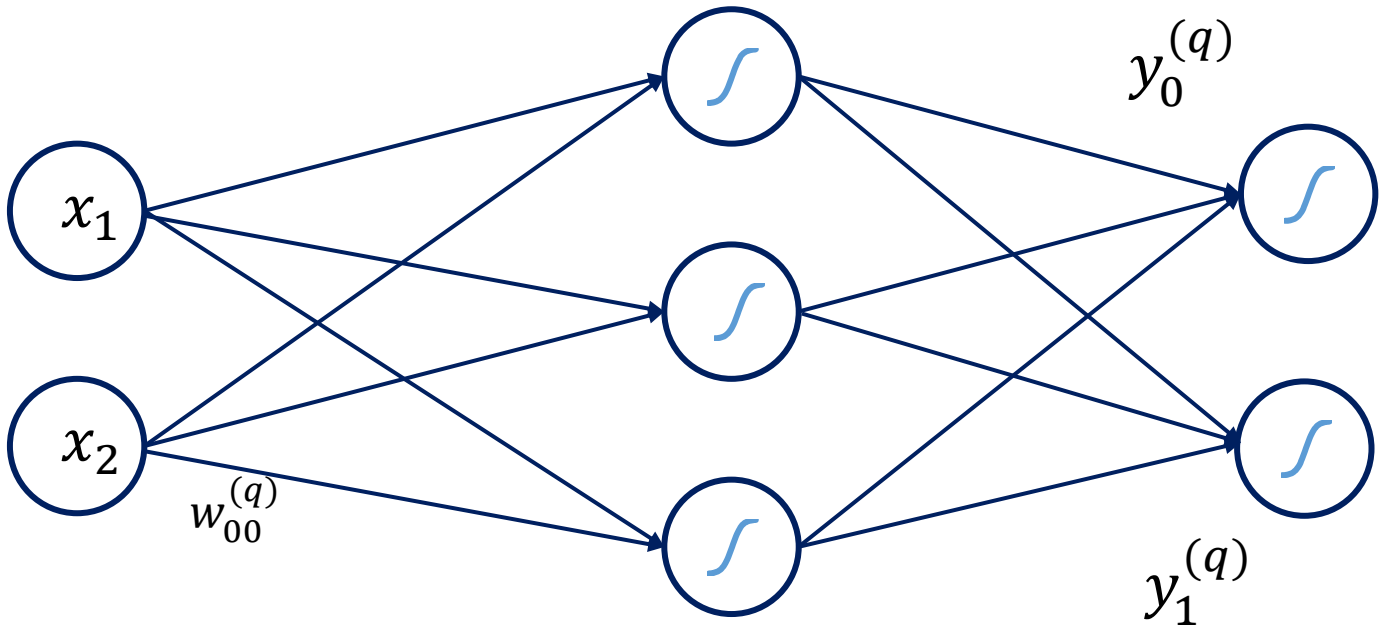
$$J = \frac{1}{2N}\sum_{i=1}^{N}\left|y_i^{(q)} - d_{(i)}\right|^2 : \qquad \frac{\partial J}{\partial y_i^{(q)}} = \left(y_j^{(q)} - d_j\right)$$

$$y_i^{(q)} = \begin{bmatrix} S_i^{(q)}, S_i^{(q)} > 0 \\ 0, S_i^{(q)} \le 0 \end{bmatrix} : \qquad \frac{dy_i^{(q)}}{dS_i^{(q)}} = \begin{bmatrix} 1, S_i^{(q)} > 0 \\ 0, S_i^{(q)} \le 0 \end{bmatrix} \qquad y_i^{(q)} = \frac{1}{1 + \exp(-S_i^{(q)})} : \qquad \frac{\partial y_i^{(q)}}{\partial S_i^{(q)}} = y \cdot (1 - y)$$

$$S_i^{(q)} = \sum_j x_j^{(q)} w_{ij}^{(q)} \qquad \frac{\partial S_i^{(q)}}{\partial w_{ij}^{(q)}} = x_j^{(q)} \qquad x_j^{(q)} = y_j^{(q-1)}$$

Обратное распространение -2

$$\frac{\partial J}{\partial w_{ij}^{(q)}} = \frac{\partial J}{\partial y_i^{(q)}} \cdot \frac{dy_i^{(q)}}{dS_i^{(q)}} \cdot \frac{\partial S_i^{(q)}}{\partial w_{ij}^{(q)}}$$



Скрытые слои:

$$\frac{\partial J}{\partial y_i^{(q)}} = \sum_k \frac{\partial J}{\partial y_k^{(q+1)}} \cdot \frac{dy_k^{(q+1)}}{dS_k^{(q+1)}} \cdot \frac{\partial S_k^{(q+1)}}{\partial y_i^{(q)}}$$

$$\frac{\partial S_k^{(q+1)}}{\partial y_i^{(q)}} = w_{ik}^{(q+1)} \qquad x_j^{(q+1)} = y_j^{(q)}$$

$$\delta_i^{(q)} = \frac{\partial J}{\partial y_i^{(q)}} \cdot \frac{dy_i^{(q)}}{dS_i^{(q)}}$$

$$\delta_i^{(q)} = \left[\sum_k \delta_k^{(q+1)} w_{ik}^{(q+1)}\right] \frac{dy_i^{(q)}}{dS_i^{(q)}}$$

Выходной слой: $\delta_i^{(q)} = \left(y_j^{(q)} - d_j\right) \frac{\partial y_i^{(q)}}{\partial S_i^{(q)}}$ *для MSE*

Правило настройки $\Delta w_{ij}^{(q)} = -\eta \delta_j^{(q)} y_i^{(q-1)}$

(метод наискорейшего спуска):

Проблемы оптимизации

- Попадание в локальный минимум
- Сложный ландшафт функции потерь: Чередование плато и сильной нелинейности
- Скорость обучения
- Отличия в скорости обновления весов, ассоциированных с разными информативными признаками

Регуляризация

L1 - регуляризация

$$J(w) = J_0(w) + \lambda\|w\|_1 \qquad L1: \lambda\|w\|_1 = \lambda\sum_{j=1}^{m}|w_j|$$

L2 - регуляризация

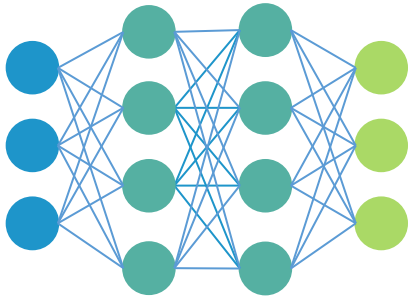
$$J(w) = J_0(w) \ + \lambda\|w\|_2^2 \qquad L2: \ \lambda\|w\|_2^2 = \lambda\sum_{j=1}^{m} w_j^2$$

Эластичная сеть

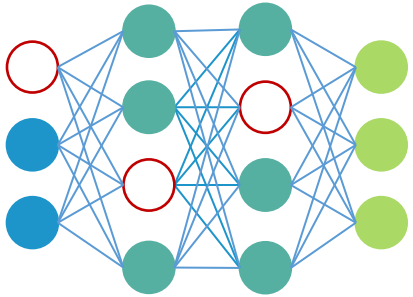
$$J(w) = J_0(w) +) + \lambda\|w\|_1 + \lambda_2\|w\|_2^2$$

$J_0(w)$ Стандартно определённая функция потерь

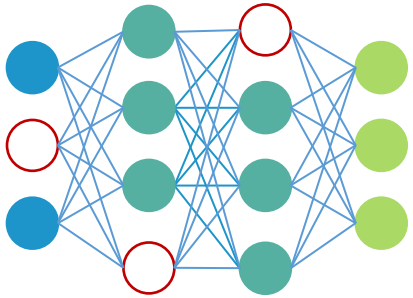
Прореживание



Все нейроны активны



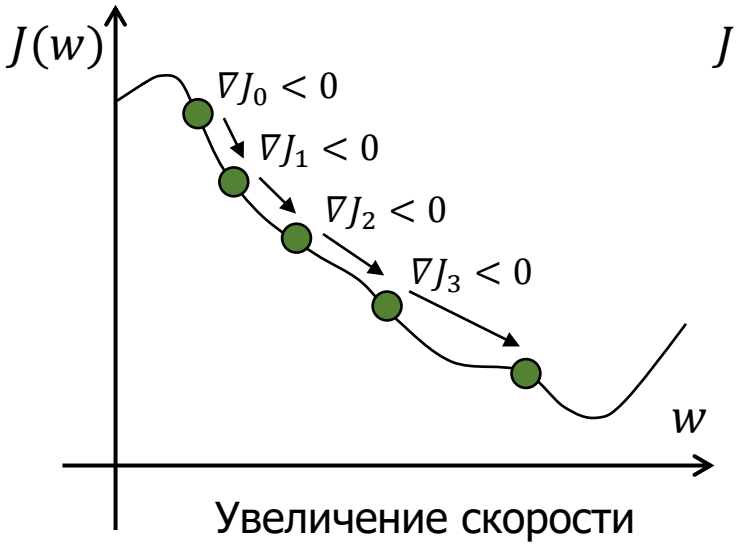
Эпоха обучения 1. Прореживание 25%



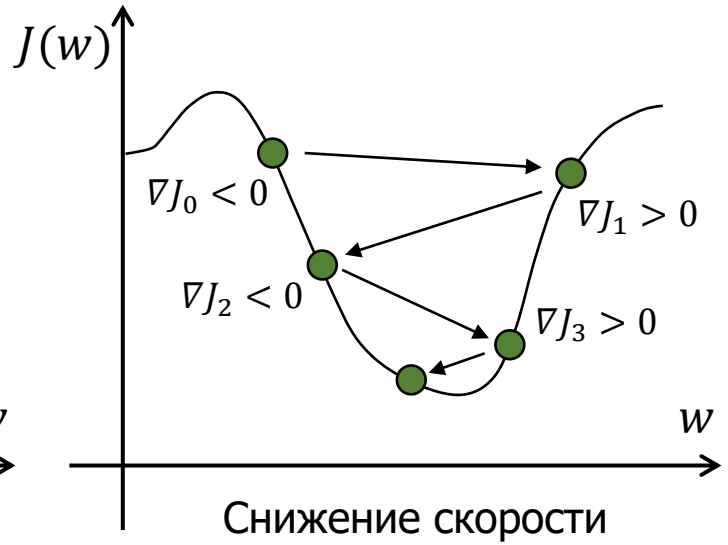
Эпоха обучения 2. Прореживание 25%

Нейроны, чей выход установлен в нуль

Обучение с учетом момента



Увеличение скорости настройки



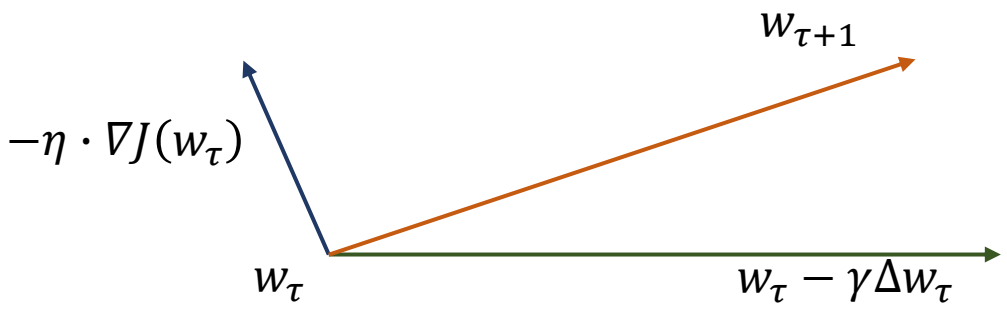
Снижение скорости настройки

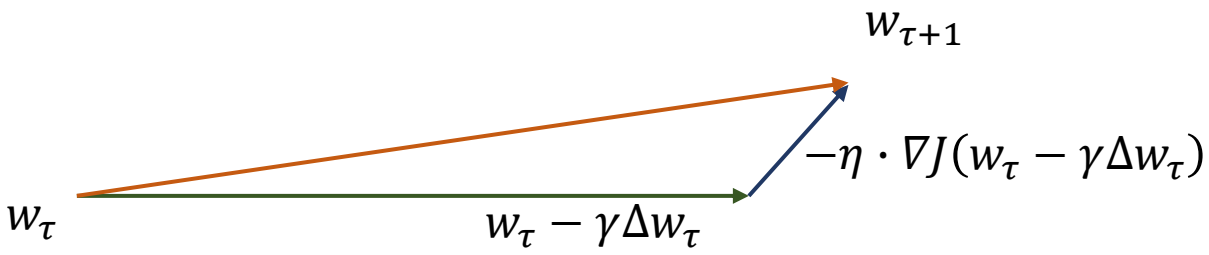
$$w_i^{(\tau+1)} = w_i^{(\tau)} - \eta^{(\tau)} \frac{\partial J}{\partial w_i^{(\tau)}} + \alpha \Delta w_i^{(\tau)}$$

$$\Delta w_i^{(\tau)} = w_i^{(\tau)} - w_i^{(\tau-1)} = \alpha \Delta w_i^{(\tau-1)} - \eta^{(\tau-1)} \frac{\partial J}{\partial w_i^{(\tau-1)}}$$

$$0 < \alpha < 1$$

Ускоренный градиент Нестерова



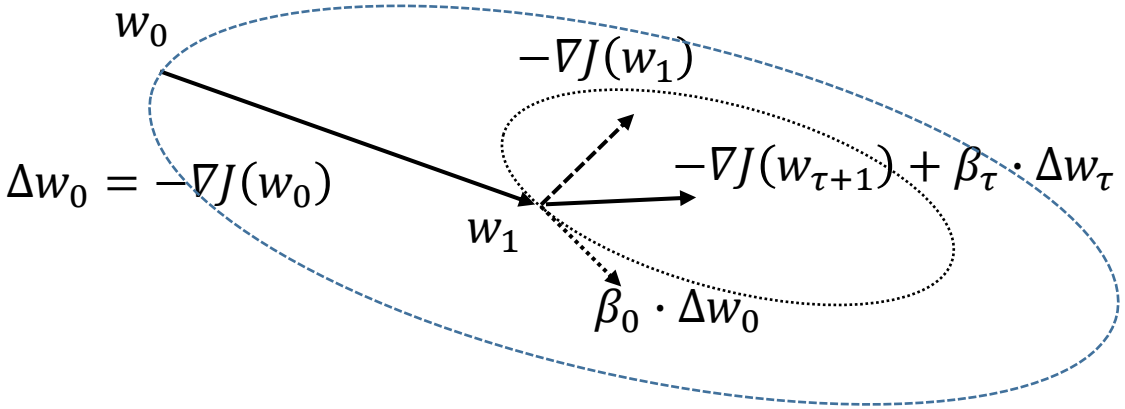


$$\Delta w_{\tau+1} = -(\gamma \Delta w_\tau + \eta \cdot \nabla J(w_\tau - \gamma \Delta w_\tau))$$

$$w_{\tau+1} = w_\tau + \Delta w_{\tau+1}$$

$\gamma \approx 0.9$ Параметр момента

Алгоритм сопряженных градиентов



$$\Delta w_{\tau+1} = -\nabla J(w_{\tau+1}) + \beta_\tau \cdot \Delta w_\tau$$

$$w_{\tau+1} = w_\tau + \Delta w_{\tau+1}$$

Коэффициент сопряжения

$$\beta_\tau = \frac{\nabla J^T(w_{\tau+1})\,(\nabla J(w_{\tau+1}) - \nabla J(w_\tau))}{\nabla J^T(w_\tau) \cdot \nabla J(w_\tau)}$$

или

$$\beta_\tau = \frac{\nabla J^T(w_{\tau+1})\,(\nabla J(w_{\tau+1}) - \nabla J(w_\tau))}{-\Delta w_\tau \cdot \nabla J(w_\tau)}$$

Алгоритм переменной метрики

$$\Delta\, w_{\tau+1} = -\eta \cdot [H(w_\tau)]^{-1} \cdot \nabla J(w_\tau)$$

Корректировка веса с учётом квадратичного приближения функции $J(w)$

$H(w_\tau)$ Гессиан

$V(w_\tau) = [G(w_\tau)]^{-1}$

$\mathrm{G}(w_\tau)$ Приближение $H(w_\tau)$

Метод Бройдена-Флетчера-Гольдфарба-Шенно (BFGS)

$$V_\tau = V_{\tau-1} + \left[1 + \frac{r_\tau^T V_{\tau-1} r_\tau}{s_\tau^T r_\tau}\right] \frac{s_\tau s_\tau^T}{s_\tau^T r_\tau} - \frac{s_\tau r_\tau^T V_{\tau-1} r_\tau s_\tau^T}{s_\tau^T r_\tau}$$

Метод Девидона-Флетчера-Пауэлла (DFP)

$$V_\tau = V_{\tau-1} + \frac{s_\tau s_\tau^T}{s_\tau^T r_\tau} - \frac{V_{\tau-1} r_\tau r_\tau^T V_{\tau-1}}{r_\tau^T V_{\tau-1} r_\tau}$$

$$s_\tau = w_\tau - w_{\tau-1} \qquad r_\tau = \nabla J(w_\tau) - \nabla J(w_{\tau-1})$$

Оптимизационные процедуры

Управление скоростью обучения:

AdaGrad

$$w_i^{(\tau+1)} = w_i^{(\tau)} - \frac{\eta^{(\tau)}}{\sqrt{\sum_{t=1}^{\tau} \frac{\partial J}{\partial (w_i^{(\tau)})} + \varepsilon}} \cdot \frac{\partial J}{\partial w_i^{(\tau)}}$$

RMSProp (Root-mean-squared propagation)

$$w_i^{(\tau+1)} = w_i^{(\tau)} - \frac{\eta^{(\tau)}}{\sqrt{m_i^{(\tau)} + \varepsilon}} \cdot \frac{\partial J}{\partial w_i^{(\tau)}} \qquad m_i^{(\tau)} = \alpha m_i^{(\tau-1)} + (1-\alpha)\frac{\partial J}{\partial w_i^{(\tau)}} \qquad 0 < \alpha < 1$$

Adam

Скользящее среднее градиента

$$m_{\tau+1} = \beta_1 m_\tau + (1-\beta_1)\cdot \nabla J(w_\tau)$$

Скользящее среднее квадратов градиентов

$$v_{\tau+1} = \beta_2 v_\tau + (1-\beta_2)\cdot (\nabla J(w_\tau))^2$$

Корректировки для улучшения решения

$$\widehat{m}_\tau = \frac{m_\tau}{1-\beta_1^\tau}$$

$$\hat{v}_\tau = \frac{v_\tau}{1-\beta_2^\tau}$$

β_1, β_2 Коэффициенты близкие к 1

Обучающее правило:

$$w_{\tau+1} = w_\tau - \frac{\eta \cdot \widehat{m}_\tau}{\sqrt{\hat{v}_\tau} + \epsilon}$$

Виды градиентного спуска

Полнопакетный градиентный спуск

Использование градиента полной потери является экстремальной версией градиентного спуска. Это неэффективно, поскольку обучающая выборка может содержать тысячи или миллионы образцов. Один шаг обновления, включающий столько вычислений, крайне неэффективен. Более того, полнопакетный градиентный спуск может привести к менее обобщаемым решениям.

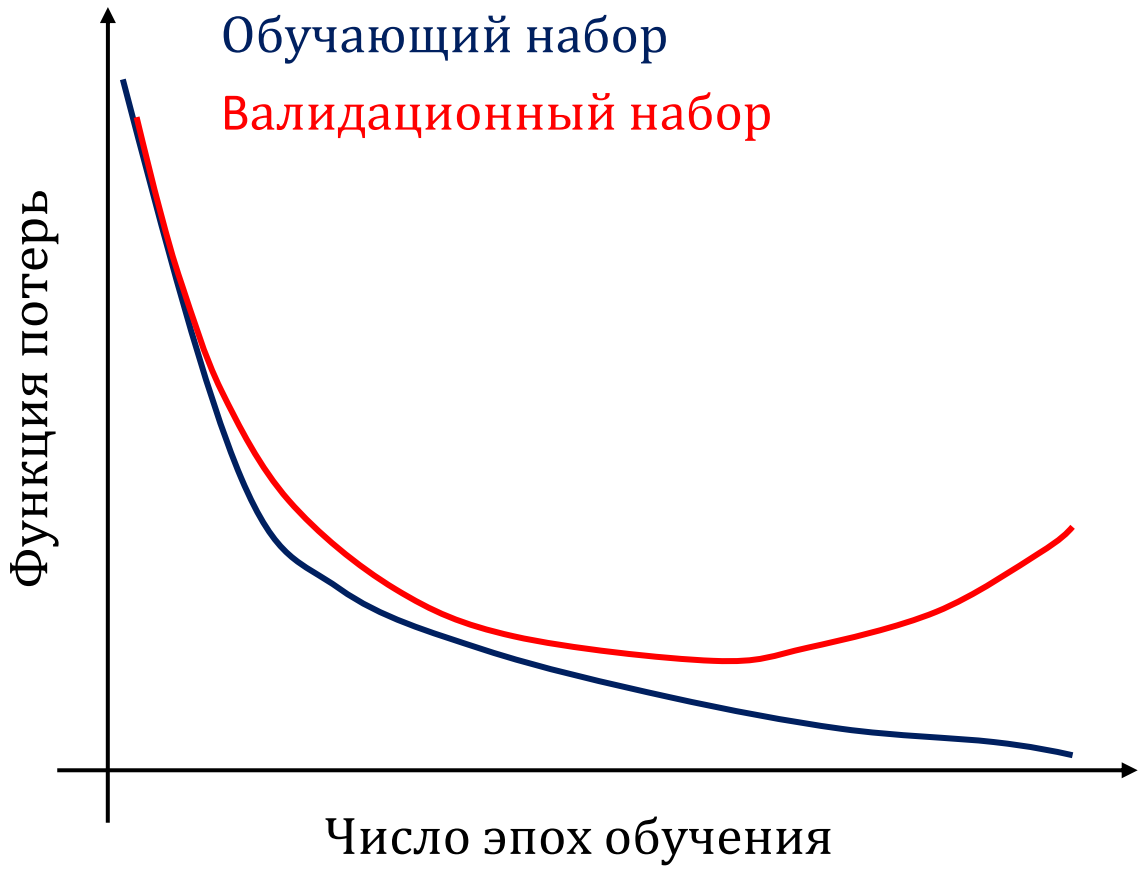
Стохастический градиентный спуск (SGD)

Стохастический градиентный спуск выбирает случайный пример за раз и обновляет параметр в соответствии с градиентом потери выборки. Этот градиент более зашумлен, чем набор градиентов, и может потребоваться больше шагов для сходимости, но он может помочь нейронным сетям избежать плохих локальных оптимумов.

Мини-пакетный градиентный спуск

Градиент для каждого обновления вычисляется для потерь, суммированных по небольшому случайному подмножеству обучающих данных. Количество выборок в подмножестве называется размером пакета. Мини-пакетная версия представляет собой компромисс вычислений для каждого обновления градиента и количества обновлений, необходимых для сходимости.

Процесс обучения



Настройка нейросети

Архитектура сети:

- Число слоёв (глубина сети)
- Число нейронов в каждом слое (ширина сети)
- Тип функции активации

Обучение и оптимизация:

- Управление скоростью обучения
- Размер пакета (mini-batch)
- Оптимизационная процедура
- Число итераций обучения

Методы регуляризации для уменьшения переобучения:

- L2 – регуляризация
- Dropout - слои
- Аугментация данных